Audit-Trail

Jede schreibende Aktion in aYOUne — Create, Update, Delete, Right-Change, Login, Customer-Switch — wird im Audit-Trail festgehalten. Für Compliance, Forensik und reine Neugier ("wer hat den Coupon entfernt?") ist das die richtige Adresse.

Was wird auditiert?

| Aktion | Erfasst | |---|---| | CRUD auf jeder Collection | Vorher/Nachher-Diff, Aktor, Zeit, IP | | Auth-Events (Login, Logout, Password-Reset) | User, IP, User-Agent, Erfolg | | Right-/Role-Changes | Was, wer, vorher !’ nachher | | Customer-Switch | Source-Customer !’ Target-Customer | | API-Token-Issue | Audience, Scope, TTL |

Audit-Records leben unter audit.ayoune.app (eigene Service-Instanz, hochskalierende Schreib-Last). Ihre TTL hängt vom Customer-Paket ab — typisch 365 Tage, für regulierte Branchen 7 Jahre.

Audit-Records einsehen (UI)

admin-v2 !’ Audit-Trail. Filter:

- Time-Range (Heute, 7d, Custom)
- Aktor (User, Service-Account, System)
- Kategorie (auth, crud, rights, ...)
- Collection (consumers, tasks, ...)
- Free-Text-Search (über Diffs)

Klick auf eine Zeile zeigt den Diff-View — alte vs. neue Werte side-by-side, plus Metadata wie IP, User-Agent, debugId.

CLI-Zugriff

```
ay audit consumers <id>           # Audit-Trail eines Datensatzes
ay history consumers <id>         # Alias

ay list audits \
  --filter '{"category":"rights","createdAt":{"$gte":"2026-04-01"}}' \
  --sort 'createdAt:-1' --limit 100
```

Output ist standardmäßig eine Tabelle. Mit --responseFormat json kannst du den Audit-Trail in dein SIEM exportieren.

Programmatischer Zugriff

Direkt-API: https://audit.ayoune.app

```
curl -X POST https://audit.ayoune.app/audits/aggregate \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "pipeline": [
      { "$match": { "category": "auth", "subcategory": "login.failed" } },
      { "$group": { "_id": "$_actorIP", "count": { "$sum": 1 } } },
```

```
      { "$sort": { "count": -1 } },
      { "$limit": 20 }
    ]
  }'
```

So findest du z.B. die Top-20 IPs mit Failed-Logins (Brute-Force-Detection).

Errors als Tasks

Eine Plattform-Eigenheit: Runtime-Errors werden zusätzlich als tasks mit type: "Bug" persistiert. Sie tauchen also nicht nur in ayounelogs (kurzlebig, ~1h TTL) sondern auch in der CRM-Task-Liste auf — gefiltert auf der Read-Seite, damit normale Sales-User sie nicht sehen.

Compliance-Export

Für DSGVO-Auskünfte oder ISO-Audits gibt es einen Bulk-Export:

```
ay exec audit:export-user --userId <id> --format csv --out ./audit-user.csv
ay exec audit:export-customer --since 2026-01-01 --format ndjson > audit-2026.ndjson
```

Der Export läuft asynchron via BullMQ-Job (audit-export) — Ergebnis wird als Download-Link gemailt.

Best Practices

- Read-Only Audit-Role für externe Auditoren — ein eigenes Login mit audit.view-Right und sonst nichts.
- Alerts auf verdächtige Patterns: 10+ Failed-Logins in 5 min !’ Slack via Notifications (wiki:admin-handbook/notifications).
- Retention-Strategie dokumentieren — was ist 365d, was 7y, warum?

Siehe auch: Error Codes (wiki:developer-portal/error-codes), Users & Roles (wiki:admin-handbook/users-and-roles).

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-10-24 exportiert. Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter /llms.txt der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com